**Comparative study of Pros and Cons of the imperative language Java and the non-imperative language Kotlin**

Jarrett Orr

University of New Orleans

New Orleans, Louisiana

jporr@uno.edu

In the paper *Comparative Study of the Pros and Cons of Programming Languages*, Java is defined as such:

A high-level programming language developed by Sun Microsystems. Java was originally called OAK, and was designed for handheld devices and set-top boxes. Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web. Java is an object-oriented language similar to C++, but simplified to eliminate language features that cause common programming errors. Java source code files (files with a .java extension) are compiled into a format called bytecode (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs), exist for most operating systems, including UNIX, the Macintosh OS, and Windows. Bytecode can also be converted directly into machine language instructions by a justin-time compiler (JIT). Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

Kotlin, on the other hand, was developed in 2011 by JetBrains. The language was formally released in 2016 but was tested thoroughly and used in small ways before that release. In 2018, Kotlin v1.3 brought coroutines for asynchronous programming. Kotlin was mainly developed to help with Java and runs on Java virtual machines, but in 2019, Google adapted Kotlin as the preferred language for Android developers. Kotlin is completely open source and supports both object-oriented and functional programming, so it is not fully non-imperative, but it is also not fully imperative. The syntax, however, is like C#, Java, and Scala. Kotlin code can run on Windows units as well as UNIX-based systems. The language is also very new, so it is a good language for a beginning programmer to learn since other programmers will be at roughly the same level.

**Kotlin vs Java**

**Extension Functions**

Kotlin allows programmers to create an extension function, but Java does not have that capability.

**Data Classes**

Java programmers spend their time writing and constructing elements to develop classes, but Kotlin does not require the same amount of work for data classes.

**Conversions**

Kotlin does not offer implicit conversions, whereas Java does offer those conversions.

**Null**

Kotlin does not have a null variable or object. Java has a null variable and object, which is used very frequently.

**Programming Types**

Java is strictly an object-oriented programming language, which means that the program is written in terms of objects instead of data and methods. Kotlin, on the other hand, is both object-oriented and functional, which means that functions are used to complete a specific task.

**Programming Specifications**

When programming in Java, one might notice the use of static members. These static members that are seen in Java do not exist in Kotlin.

In Kotlin, however, primitive variables are objects, but in Java, they are not.

Kotlin also does not require the use of semicolons when programming, but in Java, semicolons are a necessity.

The Kotlin programming language has a specification that a program can have one or more secondary conductors, but in Java secondary constructors do not exist. However, in Java, a program can have multiple constructors.

**Null Safety**

Although Kotlin does not have a null variable, Kotlin’s system has an inbuilt null safety. Java’s null safety is handled by NullPointer Exception, which also develops Android.

**Data Type Specifications**

When programming in Java, specifications are required in order to declare any variable datatype. These specifications are not required when programming in Kotlin.

**Lambda Expression**

Lambda Expression is supported by Kotlin but not by Java

**Language Scripting**

While Java does not offer language scripting capabilities, language scripting capabilities allow the programmer to use Kotlin directly in their Gradle build scripts.

**Programming Concepts**

Java supports OOPS programming concept, whereas Kotlin supports delegates, extension, higher-order functions.

**Advantages of Kotlin**

Using the Kotlin Multiplatform framework, you can extract one common codebase that will target all codebases at the same time

Kotlin offers built-in null safety support, which is a lifesaver on Android, which is full of old Java-style APIs.

Kotlin is more concise and expressive than Java, which means there is less room for mistakes.

Kotlin offers user-friendly and understandable coding norms, which help new programmers learn the language as well as help other programmers with issues since they are learning at the same time

Kotlin helps divide large apps into smaller layers, which is a very nice feature. This feature allows the programmer to not become overwhelmed with the tasks at hand since they are divided into smaller tasks.

Kotlin uses lots of function types and specialized language structures like lambda expressions, which helps programmers learn and distinguish between Kotlin and other languages. These functions also make it easier for the programmer because they can break larger problems into functions instead of having to deal with a larger problem.

Kotlin helps developers to create extension functions. This feature allows developers the opportunity that they can not find in Java and allows them to access functions from different parts of a program.

Kotlin offers a very simple and almost automated way of creating data classes. This saves the programmer a lot of time, which they would have had to spend if they were programming in Java.

Kotlin is a statically-typed language, so it is very easy to read and write. This proves again why Kotlin is a great coding language for beginners to learn since it is extremely easy to read and write.

This language allows the exchange and use of information from Java in various ways. Since most programmers have a general idea of how Java works, Kotlin allows the programmer to use information from Java and exchange those ideas from Java and use them for Kotlin code.

It’ll take less time to write new code in Kotlin. Whenever code can be written in a shorter way, that opportunity should always be taken.

It’s quite easier to deploy Kotlin code and to maintain it at scale.

**Advantages of Java**

Since Java is a widely used programming language, there are many features that have been tested and used so that mistakes can be avoided. One of those features is checked exceptions that improve error detecting and solving. This feature simply helps the programmer detect errors in their code as well as try to help solve those errors.

Detailed documentation of Java is available. This allows new programmers to catch up with some of the more experienced Java programmers as well as figure out any issues or syntax that they may be having to deal with in their Java code.

The best thing about Java is that there is a large amount of skilled developers available that can assist and help new programmers learn the more complex language of Java

There is a huge array of third-party libraries that can be used for new Java projects to help the programmer with different aspects of their code.

Java allows you to form standard programs and reusable code that make it easy for a programmer to recycle and save themselves time while programming since they have already written the code that they need.

It is a multi-threaded environment that allows you to perform many tasks simultaneously in a program. This feature allows the program to feature multiple tasks that may or may not rely on other tasks.

Java’s performance is one of the best for any programming language.

It is easy to navigate libraries, which help the programmer with certain tasks in their code.

**Disadvantages of Kotlin**

There is only a small community of developers, so learning materials and professional assistance are difficult to come by.

There is no function of checked exceptions, which may lead to errors.

Kotlin has a much slower compilation speed than Java

Kotlin is a declarative language, so sometimes it helps you to generate great amounts of boilerplate in corresponding Java Virtual Machine bytecode

**Disadvantages of Java**

Java is not suitable for Android API design because of a number of limitations.

Demands a lot of manual work which increases the number of potential errors. This proves to be an issue a lot with inexperienced programmers but can be resolved with practice.

JIT compiler makes the program comparatively slow.

Java has high memory and processing requirements. This causes an issue in terms of storage capacity, which can be resolved inexpensively by buying more storage, but it may become aggravating and cause the programmer to lose desire to program.

It does not provide support for low-level programming constructs like pointers. This is one of the reasons why Java takes a lot of memory to program.

You don’t have any control over garbage collection as Java does not offer functions like delete(), free(). The lack of garbage collection control proves detrimental to storage being another reason why Java requires a lot of memory to compile.

**Appendix**

**Java Code and Output for Fischer Random Chess**

**Text

Description automatically generated**

**Text

Description automatically generated with medium confidence**

**Kotlin Code and Output for Fischer Random Chess**

Text

Description automatically generated

Text

Description automatically generated

**References:**

**[1] Comparative study of the Pros and Cons of Programming languages**

**[2] Kotlin vs Java: What’s the Difference? By James Hartman**